Complexifying BERT using LoRA Adapters

Fabio Tamburini

Alma Mater Studiorum - Università di Bologna - Italy

fabio.tamburini@unibo.it

Introduction

- Various works proposing complex-valued Deep Neural Networks rose an increasing interest on this architectures for their intrinsic ability to manage problems defined on complex-valued features.
- For example:
  - in the fields of signal and image processing, speech, signal and audio data are naturally complex-valued after Fourier, Laplace or Complex Wavelet transforms. Yang et al. (2020) and Eilers and Jiang (2023) presented stete-of-the-art Automatic Music Transcription systems and Wang et al. (2020) evaluated their complex-valued embeddings in text classification, machine translation and language modeling with promising results.
  - Quantum-inspired Machine Learning, an emerging topic of research in NLP and AI, is completely based on complex-valued features and tensors. E.g. Liu et al. (2023) presented a survey of novel quantum-cognitively inspired models that solved the task of sentiment analysis with good performances and Tamburini (2019) proposed a Quantum WSD system.

Related Works

There are very few attempts in literature for creating a complex-valued transformer and **all of them pre-train the whole architecture from scratch**, a very long and computationally demanding process, especially for large architectures.

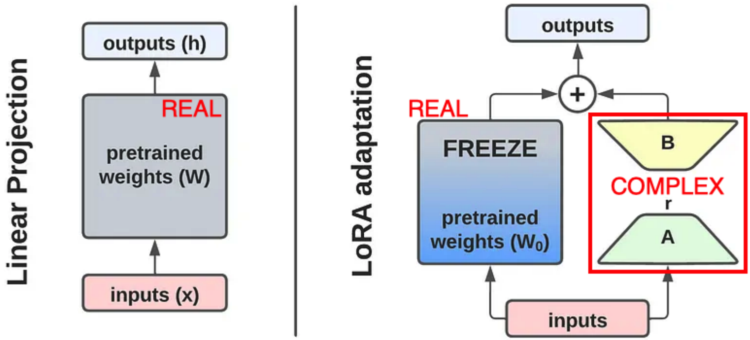
- Yang et al. (2020) concentrate on the development of a complex-valued transformer for speech, signal and audio data that are naturally complex-valued after Fourier Transform.
- Wang et al. (2020), working on positional embeddings and proposing a solution for modelling both the global absolute positions of words and their order relationships, introduced a small complex-valued transformer architecture to test their ideas.
- The works from Eilers and Jiang (2023) and Li et al. (2023) have the goal of providing a complete model for building complex-valued transformer encoders, describing possible building blocks for doing it, testing different configurations and parameters.

The General Framework

- The transformer encoder (Vaswani et al., 2017) is primarily designed for processing input text and producing intermediate representations of input sequences. It consists of multiple layers of self-attention mechanisms and feed-forward neural networks, each contributing to the encoding process of both single words and entire sequences.
- LoRA (Low-Rank Adaptation) (Hu et al., 2022) is a technique recently introduced to efficiently fine-tune transformer models. Instead of updating all the parameters of a large pre-trained model, LoRA introduces a small set of low-rank matrices, allowing the model to adapt to new tasks with significantly reduced computational and storage requirements preserving the original model's performance.
- A very recent work (Lialin et al., 2024) suggested that, by applying LoRA adapters, it is possible to **pre-train** large transformer models from scratch obtaining comparable performance with respect to regular pre-training.

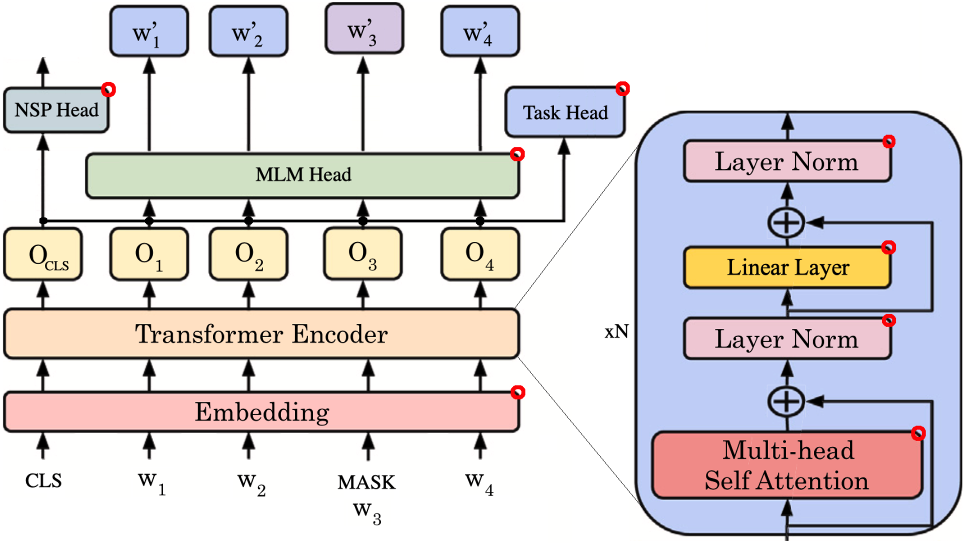
The Idea

- The main idea and contribution of this work consists in using LoRA adapters to convert a real-valued pre-trained transformer model into a complex-valued one being able to produce as output complex-valued word and sequence embeddings.



- This process requires to continue the pre-training stage of a real-valued transformer model for setting up complex-valued LoRA adapters and train the global model to produce meaningful complex-valued embeddings.

Reference Architecture: BERT



The model (1/2)

	Original 'REAL' Model	Modified 'COMPLEX' Model
Embeddings:	Embedding matrix E	Adapted by summing a complex-valued LoRA (cv-LoRA) adapter: $E' = E + A \cdot B^{\dagger}$
Linear Layers:	$Output = x \cdot W + b$, where x is the input vector, W a weight matrix and b a bias vector.	Apply a cv-LoRA adapter to the weight matrix and add a further complex-valued bias vector z: $Output = x \cdot (W + A \cdot B^{\dagger}) + (b + z)$.
Activation Function:	$GELU$	$splitGELU(z) = GELU(\mathcal{R}(z)) + i\, GELU(\mathcal{I}(z))$
Multi-head Self-Attention:	Input vector $X \in \mathbb{R}^{d \times n}$ $Q = X \cdot W^Q, K = X \cdot W^K, V = X \cdot W^V$ $Attention(Q, K, V) = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V$	Modify the three projection matrices W^Q, W^K, W^V using cv-LoRA adapters: $Attention(Q, K, V) = softmax\left(\frac{\lvert Q \cdot K^{\dagger} \rvert}{\sqrt{d_k}}\right) \cdot V$

The model (2/2)

	Original 'REAL' Model	Modified 'COMPLEX' Model
Layer Norm.:	$x' = a \frac{x-E(x)}{Var(x)} + b$	From Eilers and Jiang (2023): $$E(z) = \frac{1}{n}\sum_{j=1}^{n} z_j$$ $$Cov_{\mathbb{C}}(z) = \begin{pmatrix} Var(\mathcal{R}(z)) & Cov(\mathcal{R}(z), \mathcal{I}(z)) \\ Cov(\mathcal{R}(z), \mathcal{I}(z)) & Var(\mathcal{I}(z)) \end{pmatrix}$$ $$z' = a \cdot \sqrt{Cov_{\mathbb{C}}^{-1}(z)} \cdot \begin{pmatrix} \mathcal{R}(z - E(z)) \\ \mathcal{I}(z - E(z)) \end{pmatrix} + b$$
Training Heads & Loss Functions:	Masked Language Model (MLM), Next Sentence Prediction (NSP) and Task heads (linear projections)	cv-LoRA adapters for linear layers and Modulus function for transforming the complex-valued outputs into a real-valued one and inject it into standard loss functions.

Evaluation

Datasets

- **Continue Pre-Training.** We used the 1/3/2022 dump of the Italian Wikipedia and a "BookCorpus" we built using Italian ebooks.
- **Evaluation.** We used the UINAUIL dataset collection, a benchmark of six tasks for Italian Natural Language Understanding (Basile et al., 2023).

Task Acronym	Full name	Task type	Size (training/test)
TE	Textual Entailment	Sentence pair classification	400/400
EVENTI	Event detection & classification	Sequence labeling	5,889/917
FactA	Factuality classification	Sequence labeling	2,723/1,816
SENTIPOLC	Sentiment Polarity Classification	Sentence classification	7,410/2,000
IronITA	Irony Detection	Sentence classification	3,777/872
HaSpeeDe	Hate Speech Detection	Sentence classification	6,839/1,263

Experiments with CmplxBERTLoRA

- All the experiments rely on "*dbmdz/bert-base-italian-xxl-uncased*" (abbreviated as 'ItalianBERT_XXL') used as baseline also in Basile et al. (2023).
- During pre-training we adopted the same BERT hyperparameters, namely lr=1e-4, linear schedule with warmup and a batch size of 512.
- Our goal is to check if our complex-valued model can **produce reliable embeddings** for downstream tasks and **not to get best scores in the leaderboard**.

Model	**LoRA Rank r**	**Trainable**	**Non-Trainable**	**Total**
ItalianBERT_XXL	-	135.9M	-	135.9M
CmplxBERTLoRA_8	8	2.6M	135.9M	138.5M
CmplxBERTLoRA_16	16	5.0M	135.9M	140.9M
CmplxBERTLoRA_32	32	9.9M	135.9M	145.8M
CmplxBERTLoRA_64	64	19.7M	135.9M	155.6M
CmplxBERTLoRA_128	128	39.2M	135.9M	175.1M

Results

Model	TE				SENTIPOLC				EVENTI
	P	R	F1↑	Acc.	P	R	F1↑	Acc.	Acc.↑
Max_Freq_Baseline	.275	.500	.355	.550	.360	.500	.416	.457	.839
ItalianBERT_XXL (Basile et al., 2023)	.391	.495	.379	.541	.764	.741	.740	.675	.936
ItalianBERT_XXL (recomputed by us)	.524 ±.0608	.502 ±.0039	.383 ±.0267	.548 ±.0045	.758 ±.0051	.732 ±.0066	.733 ±.0081	.663 ±.0123	**.958** ±.0002
CmplxBERTLoRA_8	.680 ±.0548	.540 ±.0222	.453 ±.0540	.583 ±.0176	.764 ±.0107	.748 ±.0069	.747 ±.0072	.680 ±.0068	.957 ±.0006
CmplxBERTLoRA_16	.627 ±.0260	.538 ±.0166	.459 ±.0369	.580 ±.0135	.766 ±.0125	.747 ±.0059	**.750** ±.0079	.685 ±.0093	.957 ±.0003
CmplxBERTLoRA_32	.667 ±.0225	.597 ±.0698	**.551** ±.1225	.627 ±.0550	.762 ±.0065	.741 ±.0068	.742 ±.0071	.675 ±.0061	.957 ±.0012
CmplxBERTLoRA_64	.652 ±.0360	.569 ±.0528	.509 ±.0894	.606 ±.0441	.761 ±.0090	.745 ±.0102	.743 ±.0106	.674 ±.0120	**.958** ±.0007
CmplxBERTLoRA_128	.613 ±.0641	.561 ±.0555	.514 ±.0912	.592 ±.0511	.750 ±.0121	.733 ±.0107	.729 ±.0152	.657 ±.0199	.957 ±.0013

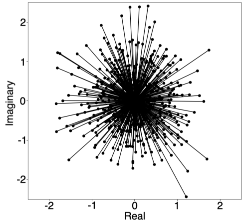
Model	IronITA				HaSpeeDe				FactA
	P	R	F1↑	Acc.	P	R	F1↑	Acc.	Acc.↑
Max_Freq_Baseline	.249	.500	.333	.499	.254	0.500	.337	.508	**.967**
ItalianBERT_XXL (Basile et al., 2023)	.769	.765	.764	.765	.792	.791	.791	.791	.908
ItalianBERT_XXL (recomputed by us)	.772 ±.0098	.769 ±.0101	**.769** ±.0102	.769 ±.0101	.790 ±.0122	.789 ±.0154	.788 ±.0165	.788 ±.0159	.911 ±.0022
CmplxBERTLoRA_8	.750 ±.0101	.746 ±.0089	.745 ±.0090	.746 ±.0089	.787 ±.0040	.784 ±.0064	.783 ±.0071	.783 ±.0066	.909 ±.0028
CmplxBERTLoRA_16	.754 ±.0075	.751 ±.0061	.751 ±.0060	.751 ±.0061	.780 ±.0076	.778 ±.0073	.777 ±.0072	.777 ±.0073	.907 ±.0028
CmplxBERTLoRA_32	.750 ±.0119	.747 ±.0095	.746 ±.0090	.747 ±.0095	.794 ±.0117	.790 ±.0132	**.789** ±.0139	.789 ±.0135	.907 ±.0022
CmplxBERTLoRA_64	.755 ±.0048	.753 ±.0040	.752 ±.0038	.753 ±.0039	.789 ±.0081	.785 ±.0106	.784 ±.0115	.784 ±.0111	.910 ±.0012
CmplxBERTLoRA_128	.744 ±.0176	.741 ±.0178	.741 ±.0180	.742 ±.0176	.785 ±.0116	.779 ±.0134	.777 ±.0142	.778 ±.0137	.909 ±.0031

Discussion

- We have to say that the UINAUIL benchmark is not without problems:
  - TE dataset is very small and such large models struggle to reliably converge to a reasonable minimum during training, leading to very unstable results.
  - FactA is very problematic as well because classes are strongly skewed and the Max_Freq_Baseline, always choosing the highest-frequency class, is able to achieve an accuracy of 96.7%!

  For all these reasons, we think that these two benchmarks should be excluded from any further evaluation.

- Are produced embeddings really non-zero and complex-valued?



Conclusions

- This pilot study presented a relevant set of experiments for testing the idea of being able to **'complexify' a Transformer encoder architecture** like BERT by using complex-valued LoRA adapters.
- The obtained results on Italian models are very encouraging showing in a clear way that this technique is effective and it maintains the same level of performance.
- This technique can in principle be used for complexifying any kind of transformer.
- A CmplxBERTLoRA model can be trained on a single 12/16GB GPU without problems, while the pre-training of a full complex-valued BERT model from scratch require at least 4 NVIDIA A100 64GB GPUs for about 500 hours!
- Thus, using LoRA for complexifying a model mitigates the need of complex and expensive computational infrastructures not easily available to any scholar.
- Code and models will be made available soon.

Time to finish!

Questions?

References

Valerio Basile, Livio Bioglio, Alessio Bosca, Cristina Bosco, and Viviana Patti. 2023. UINAUIL: A unified benchmark for Italian natural language understanding. In Danushka Bollegala, Ruihong Huang, and Alan Ritter, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Toronto, Canada, pages 348–356.

Florian Eilers and Xiaoyi Jiang. 2023. Building Blocks for a Complex-Valued Transformer Architecture. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Signal Processing Society.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*.

Qiuchi Li, Benyou Wang, Yudong Zhu, Christina Lioma, and Qun Liu. 2023. Adapting Pre-trained Language Models for Quantum Natural Language Processing.

Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. 2024. ReLoRA: High-Rank Training Through Low-Rank Updates. In *Proceedings of the International Conference on Learning Representations*. Vienna, Austria.

Yaochen Liu, Qiuchi Li, Benyou Wang, Yazhou Zhang, and Dawei Song. 2023. A survey of quantum-cognitively inspired sentiment analysis models. *ACM Comput. Surv.* 56(1).

Fabio Tamburini. 2019. A quantum-like approach to word sense disambiguation. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. INCOMA Ltd., Varna, Bulgaria, pages 1176–1185.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., volume 30.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. 2020. Encoding word order in complex embeddings. In *Proceedings of the International Conference on Learning Representations*.

Muqiao Yang, Martin Q. Ma, Dongyu Li, Yao-Hung Hubert Tsai, and Ruslan Salakhutdinov. 2020. Complex Transformer: A Framework for Modeling Complex-Valued Sequence. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*. pages 4232–4236.